逻辑回归上的反拉加速方法

The Counter Pull Acceleration Method for Logistic Regression

何沧平
微博
cangping@staff.weibo.com

2018 年 3 月 12 日

摘要

本文通过严格数学分析找出了逻辑回归过拟合的成因：边界样本的损失贡献比重大且随法向量增长而加速增大、边界样本分布散乱，顺便理清了正则项的作用机理。利用过拟合机制，本文提出一种反拉方法，既能缓解过拟合，又能减少训练步数，在 MNIST 数据集上实现加速 38.25 倍，在 CIFAR10 数据集上实现加速 5.61 倍。

In this paper, I found the two reasons of overfitting of logistic regression: boundary samples occupy a larger and larger share as the length of normal vector becomes longer and longer, boundary samples do not fit their probability density function well. With the help of insight in overfitting, I propose a acceleration method for logistic regression and got a training speedup of 38.25 on MNIST dataset, a training speedup of 5.61 on CIFAR10 dataset.

关键字: 逻辑回归，过拟合解释，反拉加速

1 引言

逻辑回归 (Logistic Regression) 是机器学习的一个基础分类方法 [1]。它形式简单，有 LIBLINEAR [2] 这样的工具库，工程实现方便，在互联网推荐系统中有广泛的应用。各大公司有成千上万台服务器在一刻不停地训练逻辑回归模型，如果能保证正确率的前提下大幅提高训练速度，那么将能节省大量运营成本。

目前提高训练速度的主要手段是样本预处理和设计更好的最优化算法。一个有效的样本预处理方法是按分量白化 [3]；可用的最优化算法有很多，常用的是梯度下降法的多种变体 [4]，例如随机梯度法、Momentum 算法、Nesterov accelerated gradient 算法、Adagrad 算法、Adadelta 算法，等等；还有 DFP、BFGS、L-BFGS 等拟牛顿算法 [6]，以及速度更快的信赖域算法 [7]；并行化的最优化算法 [5] 也能提高训练速度。

过拟合是计算学习的关键障碍，通常的解释是模型过于复杂 [1,8]，要用相对简单的模型来缓解过拟合现象；至于过拟合的成因，可用“偏差 -方差分解”[1,9] 来解释，[10] 还讨论了过拟合与噪声、多重假设检验的关系。缓解过拟合的常用手段是添加正则化项，[8] 对比了 L_1 正则化和 L_2 正则化的特点。

本文的初衷是探究逻辑回归过拟合的形成机制，因为模型已经确定，所以无法再用“模型过于复杂”这样的理由来解释。因此跳出常规的概率视角，用 Taylor 展开分析交叉熵后发现，逻辑回归过拟合的原因有两个：*边界样本的损失贡献比重大且随法向量增长而加速增大、边界样本分布散乱*。虽然法向量过大只是过拟合的表象，但是控制法向量模长却能够切实缓解过拟合，因此各种正则化手段有效。

利用对过拟合机制的洞察，本文提出一种反拉方法：修改各个样本在交叉熵损失函数中的贡献比重，提高被分错样本的损失贡献，能够减少提高逻辑回归的训练次数；降低被分错样本的损失贡献，能够减缓过拟合。为了保证交叉熵的数值稳定性，顺便提出一种近似计算方法。在手写数字数据集 MNIST [12] 上，反拉方法将训练速度提高 38.25 倍；在 CIFRA10 数据集上，反拉方法将训练速度提高 5.61 倍。

本文后续内容这样组织。第 2 节给出逻辑回归公式，为后文公式推导做准备；第 3 节给合实例和公式推导给出过拟合的 2 个原因；第 4 节给出反拉方法；第 5 节是数值实验，验证反拉方法的加速性能和缓解过拟合的效果。

2 逻辑回归

给定数据集 $D=\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\ldots,(\mathbf{x}_m,y_m)\}$, 其中 d 为正整数，列向量 $\mathbf{x}_i=(x_{i1};x_{i2};\ldots;x_{id})$，标量 $y_i\in\{0,1\}$。当 $y_i=0$ 时称 $\mathbf{x}_i$ 是负样本，当 $y_i=1$ 时称 $\mathbf{x}_i$ 是正样本。二分类问题是要从数据集 D 中学习到一个模型，然后用这个模型预测任意的样本 $\mathbf{x}_j$ 是正样本还是负样本。

逻辑回归的任务是从给定数据集 D 中学习到*分隔面*的斜截式方程

$$\mathbf{w}^T\mathbf{x}+b=0,\quad |\mathbf{w}|\neq 0 \tag{1}$$

确定其中的法向量 $\mathbf{w}$ 和截距 b 值。这里的 $\mathbf{w}$ 是列向量，记为 $\mathbf{w}=(w_1;w_2;\ldots;w_d)$，$b$ 是标量。任意平面都可以用做分隔面，区别只是推测效果可能不同。

为了寻找分隔面 (1)，对 $\forall\mathbf{x}_i\in D$，令 $z_i=\mathbf{w}^T\mathbf{x}_i+b$，按照 [11] 中定义，$z_i$ 为点 $\mathbf{x}_i$ 到分隔面 (1) 的*加权距离*。定义单个样本 $\mathbf{x}_i$ 上的损失函数

$$l(z_i)=\begin{cases}-\ln(1-\sigma(z_i)), & \text{如果 } y_i=0,\\ -\ln(\sigma(z_i)), & \text{如果 } y_i=1,\end{cases} \tag{2}$$

这里的 $\sigma(z)$ 为 Sigmoid 函数 $\sigma(z)=\frac{1}{1+e^{-z}}$。将样本集 D 上的损失函数定义为

$$L(\mathbf{w},b)=\frac{1}{m}\sum_{i=1}^{m}l(z_i),$$

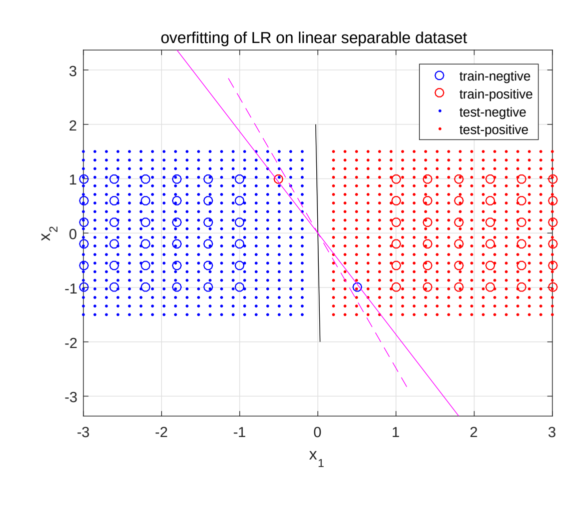


图 1: 线性可分样本集上的过拟合

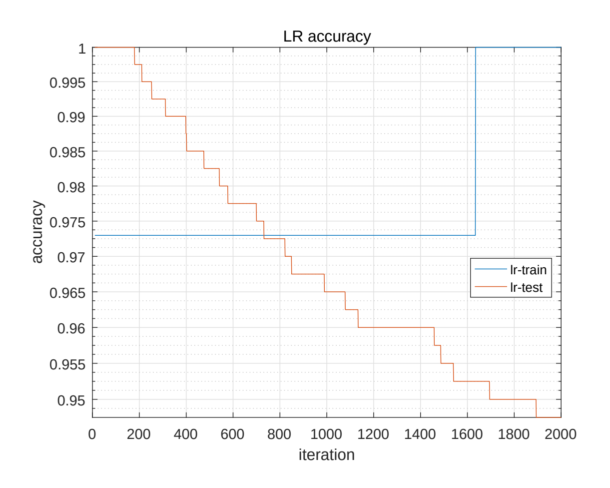


图 2: 线性可分样本集上的正确率

求解它的最小值

$$\{\hat{\mathbf{w}}, \hat{b}\} = \arg\min_{\mathbf{w},b} L(\mathbf{w}, b), \tag{3}$$

就得到了最优参数 $\hat{\mathbf{w}}$ 和 $\hat{b}$，代入 (1) 即得最优分隔面 $\hat{\mathbf{w}}^T\mathbf{x} + \hat{b} = 0$。

对任意样本 $\mathbf{x}_j$，用最优分隔面来推测它归属的类别。令

$$y_j = \begin{cases} 0, & \text{如果 } \hat{\mathbf{w}}^T\mathbf{x}_j + \hat{b} < 0, \\ 1, & \text{如果 } \hat{\mathbf{w}}^T\mathbf{x}_j + \hat{b} \geq 0, \end{cases} \tag{4}$$

如果 $y_j = 0$，那么推测 $\mathbf{x}$ 是负样本；如果 $y_j = 1$，那么推测 $\mathbf{x}_j$ 是正样本。

3 过拟合实例与成因

在逻辑回归问题中，正确率通常会随着训练步数的增加而升高。有时在训练若干步以后，随着训练集样本上的正确率逐渐提高，测试集上的正确率不再提高甚至下降，这种现象称为过拟合。

为直观说明过拟合的成因，先给出 2 个没有实际意义的例子，它们分别对应线性可分的样本集和线性不可分的样本集。

3.1 线性可分样本集上的过拟合

图 1中，蓝色圆圈是训练集中的负样本，红色圆圈是训练集中的正样本。训练集中的 36 个负样本均匀分布在区域 $[-3, -1] \times [-1, 1]$ 中，一个偏离主体的训练集负样本是点 (0.5,-1)。训练集中的 36 个正样本均匀分布在区域 $[1, 3] \times [-1, 1]$ 中，一个偏离主体的训练集正样本是点 (-0.5,1)。根据 [11] 中的定义，可以验证这个训练集线性可分。20×20 个蓝色小圆点是测试集中的负样本，它

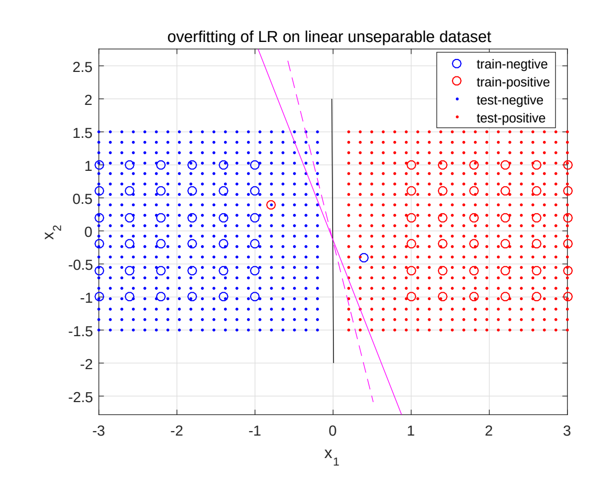


图 3: 线性不可分样本集上的过拟合

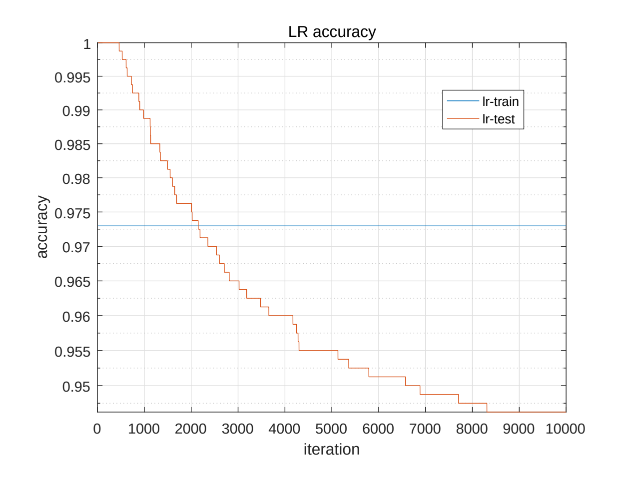


图 4: 线性不可分样本集上的正确率

们均匀分布在区间 $[-3,-0.2]\times[-1.5,1.5]$ 中；20×20 个红点小圆点是测试集中的正样本，它们均匀分布在区间 $[0.2,3]\times[-1.5,1.5]$ 中。

使用逻辑回归对这个样本集分类，用最速下降法迭代求解式 (3)，迭代步长指定为 0.1。图 1中的黑色直线是初始分隔线 (分隔面在二维空间退化为分隔线)，洋红色虚线是迭代 1000 步后的分隔线，洋红色实直线是迭代 2000 步后的分隔线。黑色直线是按照 [11] 中的方法选取的：

$$\frac{(\boldsymbol{\mu}_0+\boldsymbol{\mu}_1)^T}{|\boldsymbol{\mu}_0+\boldsymbol{\mu}_1|}(\mathbf{x}-\frac{\boldsymbol{\mu}_0+\boldsymbol{\mu}_1}{2})=0, \tag{5}$$

这里的 $\boldsymbol{\mu}_0$ 是训练集中所有负样本的均值，$\boldsymbol{\mu}_1$ 是训练集中所有正样本的均值。

图 2是迭代过程中的正确率走势，在第 1635 步迭代之后，训练集上的正确率达到了 1，但测试集上的正确率从第 180 步开始持续下降，发生过拟合。

3.2 线性不可分样本集上的过拟合

图 3中，蓝色圆圈是训练集中的负样本，红色圆圈是训练集中的正样本。训练集中的 36 个负样本均匀分布在区域 $[-3,-1]\times[-1,1]$ 中，一个偏离主体的训练集负样本是点 (0.4,-0.4)。训练集中的 36 个正样本均匀分布在区域 $[1,3]\times[-1,1]$ 中，一个偏离主体的训练集正样本是点 (-0.8,0.4)。根据 [11] 中的定义，这个训练集线性不可分。20×20 个蓝色小圆点是测试集中的负样本，它们均匀分布在区间 $[-3,-0.2]\times[-1.5,1.5]$ 中；20×20 个红点小圆点是测试集中的正样本，它们均匀分布在区间 $[0.2,3]\times[-1.5,1.5]$ 中。

使用逻辑回归对这个样本集分类，用最速下降法迭代求解式 (3)，迭代步长指定为 0.1。图 3中的黑色直线是初始分隔线，洋红色虚线是迭代 5000 步后的分隔线，洋红色实直线是迭代 10000 步后的分隔线。黑色直线的方程是 (5)。图 4是迭代过程中的正确率走势，训练集上的正确率保持平稳，但测试集上的正确率从 463 步开始持续下降，发生过拟合。

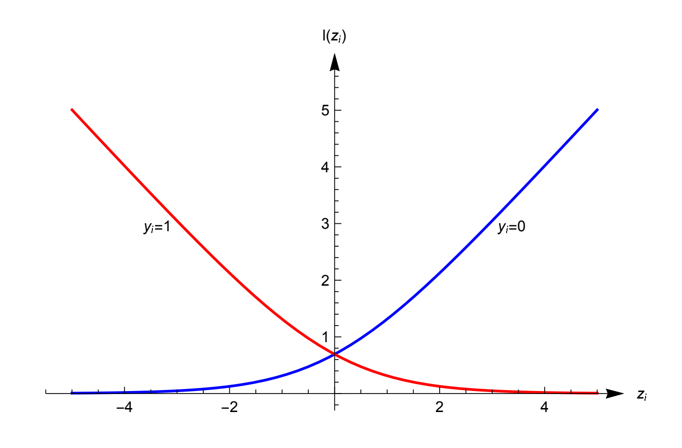


图 5: 单个样本上的损失函数 $l(z_i)$。红线对应正样本，蓝线对应负样本。

仔细考查图 1和图 3发现，很少的边界样本的大致决定了分隔面的走向，边界样本的影响力比远离边界的样本的影响力大很多，这也许就是探寻过拟合线索。

3.3 过拟合成因

人眼直观判断，图 1和图 3中各有 2 个训练样本远离主体，应该按噪音处理，舍去；即使不舍去，它们对确定分隔线的影响也不应太大。实际上，如果舍去噪音样本，那么训练集得到的理想分隔线应该为 $x_1 = 0$。黑色直线方程为 $0.9999x_1 + 0.0140x_2 = 0$，与理想分割线很接近。

逻辑回归得到的分隔线是怎么偏离样本主体的呢？为此，仔细观察损失函数 $l(z_i)$ 的走势。样本 $\mathbf{x}_i$ 与 z_i 一一对应，从图 5中知道，对正样本 $\mathbf{x}_i$，如果 $z_i \geq 0$，那么 $\mathbf{x}_i$ 被正确分类，此时它的损失函数值 $l(\sigma(z_i)) \leq -\ln(\sigma(0))$；如果 $z_i < 0$，那么 $\mathbf{x}_i$ 被错误地分为负类，此时它的损失函数值 $l(\sigma(z_i)) > -\ln(\sigma(0))$。当 $\mathbf{x}_i$ 为负样本时，情况类似。

从图 5中可以直观地看到，相对于被正确分类的样本，被错误分类的样本对损失函数的贡献更大。

为了定量分析样本对损失函数的贡献，需要用 Taylor 公式寻找 $l(z_i)$ 的简单近似函数。为此定义两个函数

$$f_0(z) = \begin{cases} -e^z, & \text{如果} z < -C_0 < 0, \\ \ln(1-\sigma(z)), & \text{如果} -C_0 \leq z \leq C_0, \\ -z - e^{-z}, & \text{如果} z > C_0 > 0, \end{cases} \tag{6}$$

$$f_1(z) = \begin{cases} z - e^z, & \text{如果} z < -C_0 < 0, \\ \ln(\sigma(z)), & \text{如果} -C_0 \leq z \leq C_0, \\ -e^{-z}, & \text{如果} z > C_0 > 0, \end{cases} \tag{7}$$

这里的 C_0 是任意指定的正实数。

定理 1. 函数 $f_0(z)$ 是 $\ln(1-\sigma(z))$ 的近似，函数 $f_1(z)$ 是 $\ln(\sigma(z))$ 的近似。

证. 先证明 $f_0(z)$ 是 $\ln(1-\sigma(z))$ 的近似。当 $z < -C_0$ 时，$e^z < \exp(-C_0) < 1$，从而有

$$\begin{aligned}\ln(1-\sigma(z)) &= \ln(1-\frac{1}{1+e^{-z}}) = \ln(1-\frac{e^z}{1+e^z}) = \ln(\frac{1}{1+e^z}) = \ln(1) - \ln(1+e^z) \\ &= -e^z + O(e^{2z}).\end{aligned}$$

当 $z > C_0$ 时，$e^{-z} < \exp(-C_0) < 1$，从而有

$$\begin{aligned}\ln(1-\sigma(z)) &= \ln(1-\frac{1}{1+e^{-z}}) = \ln(\frac{e^{-z}}{1+e^{-z}}) = \ln(e^{-z}) - \ln(1+e^{-z)} \\ &= -z - e^{-z} + O(e^{-2z}).\end{aligned}$$

因此，对任意给定的实数 z，$\max(|f_0(z)-\ln(1-\sigma(z))|) = O(\exp(-2C_0))$，函数 $f_0(z)$ 是 $\ln(1-\sigma(z))$ 的近似。

再证明 $f_1(z)$ 是 $\ln(\sigma(z))$ 的近似。当 $z < -C_0$ 时，$e^z < \exp(-C_0) < 1$，从而有

$$\begin{aligned}\ln(\sigma(z)) &= \ln(\frac{1}{1+e^{-z}}) = \ln(\frac{e^z}{1+e^z}) = \ln(e^z) - \ln(1+e^z) \\ &= z - e^z + O(e^{2z}).\end{aligned}$$

当 $z > C_0$ 时，$e^{-z} < \exp(-C_0) < 1$，从而有

$$\ln(\sigma(z)) = \ln(\frac{1}{1+e^{-z}}) = \ln(1) - \ln(1+e^{-z}) = -e^{-z} + O(e^{-2z}).$$

因此，对任意给定的实数 z，$\max(|f_1(z) - \ln(\sigma(z))|) = O(\exp(-2C_0))$，函数 $f_1(z)$ 是 $\ln(\sigma(z))$ 的近似。

[证毕]

当 $C_0 = 4.3$ 时，$\exp(-C_0) = 0.0136$, $\exp(-2C_0) = 0.00018411$。实际上，容易验证，此时有 $0 < f_0(z) - \ln(1-\sigma(z)) < 0.0001$, $0 < f_1(z) - \ln(\sigma(z)) < 0.0001$，逼近良好。

根据定理 1，单个样本上的损失函数 (2) 可以近似地表示为

$$l(z_i) \approx \begin{cases} -f_0(z_i), & \text{如果 } y_i = 0, \\ -f_1(z_i), & \text{如果 } y_i = 1. \end{cases}$$

为简化说明，本节后续叙述只考虑正样本的损失函数曲线，负样本的情形类似。式 (1) 是分隔面的斜截式方程，由解析几何知道，它有一个等价的点法式方程 $\mathbf{w}^T(\mathbf{x}_i - \mathbf{c}) = 0$，这里的 $\mathbf{c}$ 是 d 维列向量。假设样本 $\mathbf{x}_1$ 和 $\mathbf{x}_2$ 均为正样本，给定法向量 $\mathbf{w}$ 和点 $\mathbf{c}$，有 $z_1 = \mathbf{w}^T(\mathbf{x}_1 - \mathbf{c})$ 和 $z_2 = \mathbf{w}^T(\mathbf{x}_2 - \mathbf{c})$。观察图 5中红线知道，如果 $z_1 < z_2$，那么 $l(z_i) > l(z_2)$，即

推论 1. 样本的加权距离越小，损失贡献越大。

给定 $C_0>0$。当 $z_1<z_2<-C_0$ 时，$\mathbf{x}_1$ 和 $\mathbf{x}_2$ 均位于分隔面 $\mathbf{w}^T(\mathbf{x}_i-\mathbf{c})=0$ 的背面，即都被分错了。假设 $\mathbf{w}_2=s\mathbf{w}$，其中实数 $s>1$，记 $\bar{z}_1=\mathbf{w}_2^T(\mathbf{x}_1-\mathbf{c})$ 和 $\bar{z}_2=\mathbf{w}_2^T(\mathbf{x}_2-\mathbf{c})$，那么有

$$\frac{l(\bar{z}_1)}{l(\bar{z}_2)}=\frac{l(sz_1)}{l(sz_2)}\approx\frac{-f_1(sz_1)}{-f_1(sz_2)}\approx\frac{sz_1-e^{sz_1}}{sz_2-e^{sz_2}}\approx\frac{z_1}{z_2}\approx\frac{l(z_1)}{l(z_2)}, \tag{8}$$

由式 (8) 得

推论 2. 被分错样本之间的损失贡献比例不随法向量的变化而变化。

给定 $C_0>0$。当 $C_0<z_1<z_2$ 时，$\mathbf{x}_1$ 和 $\mathbf{x}_2$ 均位于分隔面 $\mathbf{w}^T(\mathbf{x}_i-\mathbf{c})=0$ 的正面，即都被分对了。假设 $\mathbf{w}_2=s\mathbf{w}$，其中实数 $s>1$，记 $\bar{z}_1=\mathbf{w}_2^T(\mathbf{x}_1-\mathbf{c})$ 和 $\bar{z}_2=\mathbf{w}_2^T(\mathbf{x}_2-\mathbf{c})$，那么有

$$\frac{l(\bar{z}_1)}{l(\bar{z}_2)}=\frac{l(sz_1)}{l(sz_2)}\approx\frac{-f_1(sz_1)}{-f_1(sz_2)}\approx\frac{e^{-sz_1}}{e^{-sz_2}}=\exp(s(z_2-z_1))=(\exp(z_2-z_1))^s, \tag{9}$$

由式 (9) 得

推论 3. 被分对样本之间的损失贡献比例会随着法向量的增长而指数级增长。

给定 $C_0>0$。当 $z_2>C_0$ 且 $z_1=-z_2$ 时，$\mathbf{x}_1$ 和 $\mathbf{x}_2$ 分别位于分隔面 $\mathbf{w}^T(\mathbf{x}_i-\mathbf{c})=0$ 的背面和正面，即一个被分错了另一个被分对了。假设 $\mathbf{w}_2=s\mathbf{w}$，其中实数 $s>1$，记 $\bar{z}_1=\mathbf{w}_2^T(\mathbf{x}_1-\mathbf{c})$ 和 $\bar{z}_2=\mathbf{w}_2^T(\mathbf{x}_2-\mathbf{c})$，那么有

$$\frac{l(\bar{z}_1)}{l(\bar{z}_2)}=\frac{l(-sz_2)}{l(sz_2)}\approx\frac{-f_1(-sz_2)}{-f_1(sz_2)}\approx\frac{sz_2+\exp(-sz_2)}{\exp(-sz_2)}=1+sz_2\exp(sz_2), \tag{10}$$

由式 (10) 得

推论 4. 被分错样本与被分对样本之间的损失贡献比例会随着法向量的增长而指数级增长。

将分隔面附近样本称为边界样本。从推论 1~ 推论 4可知，对损失函数的贡献比例，由大到小分顺序是：被分错的样本、被分对的边界样本、被分对的其它样本，它们之间的比例关系随着法向量的增长而迅速增大。适用逻辑回归的数据集，被最优分隔面分错的样本占比不大，这样被分错的样本通常会在分隔面附近。考虑到，在线性可分数据集上，法向量模长 $|\mathbf{w}|$ 趋向无穷大 [11]，分隔平面几乎完全由边界样本决定。在线性不可分数据集上，法向量模长 $|\mathbf{w}|$ 有界 [11]，但最优分隔面的法向量模长可能仍然很大，过拟合仍然严重。因此得出过拟合原因之一：边界样本的损失贡献比重大且随权重增长而加速增大。

自然界很多事件服从正态分布，例如图 6，中心处样本密度大，能够很好在逼近其概率密度函数；在远离中心的边缘处，概率密度函数的值较小，样本稀疏，不能很好地反映其概率密度函数。考虑到训练集边界样本基本决定分隔平面，而测试集样本的实际分布与训练集会有一些差异，所以得到的分隔平面不能很好地分隔训练集。因此得到过拟合的原因之二：边界样本分布散乱。

第 3节的 2 个过拟合例子都是根据这 2 个原因设计出来的。

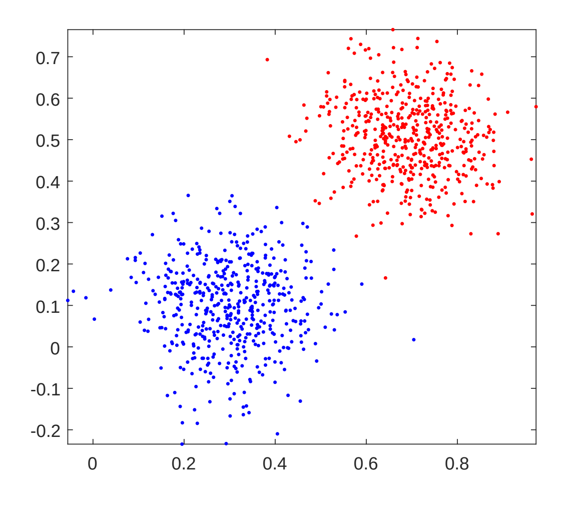


图 6: 一个服从正态分布的样本集

3.4 正则化的作用机理

缓解过拟合的常用手段是添加正则化项，各种各样的正则化方法的目标都是一致的：控制法向量的模长，不让 $|\mathbf{w}|$ 过大。由过拟合的成因可知，虽然法向量过大只是过拟合的表象，不是根本原因，但限制它的模长确实有效缓解了过拟合，这是因为它限制了边缘样本的损失贡献比重。正则化缓解过拟的同时，必然会降低训练集上的正确率。

从过拟合成因还可以知道缓解过拟合的另一个思路：修整边界样本使之准确反映概率密度函数。教科书 [1] 中已经写明增加样本数量能缓解过拟合，其实也可以用边界样本散乱的观点来解释：增加样本总量，边界样本数量也同比例增加，从而边界样本更好地反映其概率密度函数，从而缓解过拟合。

4 反拉加速

图 5画出了单个正样本 (红色) 和单个负样本（蓝色）的损失曲线。直观地理解，如果样本集是线性可分的，那么正样本对应的 z_i 越大，该样本上的损失函数值越小；负样本对应的 z_i 越小，该样本上的损失函数值越小。从而，在式 (3) 的计算过程中，负样本向 z_i 负无穷方向移动，正样本向 z_i 正无穷方向移动，达到了分类的目的。从过拟合成因的分析过程可知，对给定的 $\mathbf{w}$ 和 b，被错误分类的样本的损失贡献比重大，从而能优先减少分错样本的数量。

为了更快速地找到最优分隔面，索性进一步提高被分错样本的损失贡献，让错误更猛烈一些。

4.1 反拉方法

定义半正定 (POsitive Semidefinite) 函数

$$\mathrm{pos}(z) = \begin{cases} 0, & \text{如果 } z < 0, \\ z, & \text{如果 } z \geq 0, \end{cases}$$

和半负定 (NEgtive Semidefinite) 函数

$$\mathrm{nes}(z) = \begin{cases} z, & \text{如果 } z \leq 0, \\ 0, & \text{如果 } z > 0. \end{cases}$$

加权距离的计算方法保持不变，即

$$z_i = \mathbf{w}^T \mathbf{x}_i + b.$$

对加权距离进行反拉变换，得到

$$Z_i = \begin{cases} z_i + \lambda\, \mathrm{pos}(z_i), & \text{如果 } y_i = 0, \\ z_i + \lambda\, \mathrm{nes}(z_i), & \text{如果 } y_i = 1, \end{cases}$$

这里的 λ 称为反拉系数，取值范围是 $(-1, +\infty)$。将损失函数 (2) 替换为

$$h(z_i) = \begin{cases} -\ln(1 - \sigma(Z_i)), & \text{如果 } y_i = 0, \\ -\ln(\sigma(Z_i)), & \text{如果 } y_i = 1, \end{cases}$$

从而样本集 D 上的损失函数为

$$H(\mathbf{w}, b) = \frac{1}{m} \sum_{i=1}^{m} h(z_i).$$

求解它的最小值

$$\{\hat{\mathbf{w}}, \hat{b}\} = \arg\min_{\mathbf{w}, b} H(\mathbf{w}, b), \tag{11}$$

就得到了最优参数 $\hat{\mathbf{w}}$ 和 $\hat{b}$。

由式 (4.1)(4.1) 知，在 $\lambda = 0$ 时，$h(z_i) = l(z_i)$。图 7中对比了逻辑回归损失函数 $l(z)$ 和反拉后的损失函数 $h(z)$，可以看到，当 $\lambda > 0$ 时，对被分错的样本，反拉后的损失更大了；当 $-1 < \lambda < 0$ 时，对被分错的样本，反拉后的损失变小了一些，可以缓解过拟合。实际应用时，反拉系数 λ 的选取需要多次试探，以便找到最优值。

损失函数 $h(z)$ 的导数也容易求得

$$\frac{\partial h(z_i)}{\partial \mathbf{w}} = \begin{cases} \sigma(Z_i)[1 + \lambda\, \mathrm{pos}'(z_i)]\mathbf{x}_i, & \text{如果 } y_i = 0, \\ [\sigma(Z_i) - 1][1 + \lambda\, \mathrm{nes}'(z_i)]\mathbf{x}_i, & \text{如果 } y_i = 1, \end{cases}$$

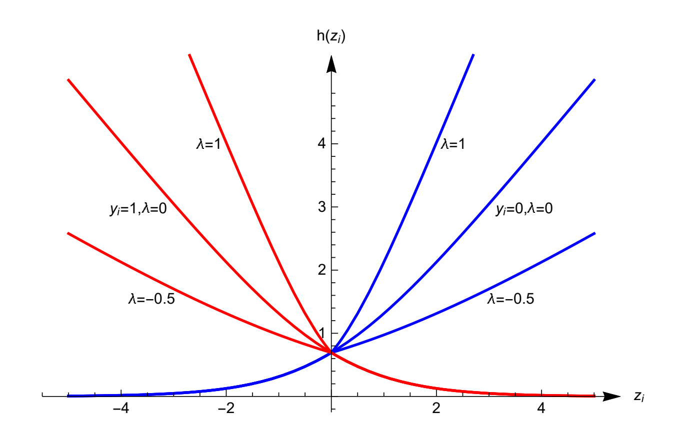


图 7: 反拉后的损失函数，蓝线对应负样本，红线对应正样本。$\lambda = 0$ 时 $h(z_i)$ 就退化为 $l(z_i)$。

$$\frac{\partial h(z_i)}{\partial b} = \begin{cases} \sigma(Z_i)[1 + \lambda\, \mathrm{pos}'(z_i)], & \text{如果 } y_i = 0, \\ [\sigma(Z_i) - 1][1 + \lambda\, \mathrm{nes}'(z_i)], & \text{如果 } y_i = 1, \end{cases}$$

这里的半正定函数的导数为

$$\mathrm{pos}'(z) = \begin{cases} 0, & \text{如果 } z < 0, \\ 1, & \text{如果 } z \geq 0. \end{cases}$$

半负定函数的导数为

$$\mathrm{nes}'(z) = \begin{cases} 1, & \text{如果 } z \leq 0, \\ 0, & \text{如果 } z > 0. \end{cases}$$

反拉加速只用于训练，不用于推测。一旦得到最优分隔面参数 $\hat{\mathbf{w}}$ 和 $\hat{b}$，仍然使用式 (4) 来推测样本的类别。

4.2 损失函数的数值稳定性

使用反拉方法后，对给定的正样本 $\mathbf{x}_i$、法向量 $\mathbf{w}$ 和截距 b，如果 $z_i < 0$，那么 λ 越大 $\sigma(Z_i)$ 越接近于 0，损失函数 $\ln(\sigma(Z_i))$ 的数值计算越不稳定，很容易超出计算机的表示范围，得到结果 NaN(Not A Number)。负样本的情形类似。

为了保持数值稳定，同时减少一点计算量，用式 (6) 近似计算式 (4.1) 中的 $\ln(1 - \sigma(Z_i))$，用式 (7) 近似计算式 (4.1) 中的 $\ln(\sigma(Z_i))$。式 (6)(4.1) 中的常数 C_0 可以根据精度要求取值，例如 $C_0 = 4.3$ 时，近似值与精确值之间的误差小于 0.0001.

5 数值实验

反拉方法的设计目标是减少迭代次数，降低训练成本，额外收获是能够缓解过拟合。

反拉方法的本质是调整了各个样本的损失比重，不涉及正则项和最优化算法，因此只需要在最优化算法、正则项相同的情况下对比逻辑回归在使用反拉方法前后的性能。

MNIST 数据集和 CIFAR10 数据集分别包含了 10 类样本，恰好可以任取 2 类样本组合起来测试反拉方法的性能。

5.1 加速 MNIST 训练

手写数字数据集 MNIST [12] 包含 0-9 这个 10 个数字的图片, 图片大小为 28×28，将 2 维单色图像拉平制作为 1 维向量。取任意两个数字的图片分别作为负样本、正样本进行训练，组合顺序依次为 0-1、0-2、. . .、0-9、1-2、1-3、. . .、1-9、. . .、7-8、8-9，一共 45 种组合。每种组合训练 10 次，然后训练下一种组合，共计训练 450 次。训练使用负梯度下降法，步长指定为 0.01，无正则化项，$\mathbf{w}$ 的初值从均匀分布 $U(-1/\sqrt{784}, 1/\sqrt{784})$ 中随机选取，b 的初值为 0。最大迭代步数设为 20000，LR 最大正确率对应的迭代步数（称为 *LR 最优迭代步数*），FLR 迭代时到达 LR 最大正确率花费的迭代步数称为 *F*LR 最优迭代步数，如图 8如示，用 LR 的最优迭代步数除以 FLR 的最优迭代步数就得到 *加速倍数*，如图 9所示。加速倍数为 1 意味着没有加速，加速倍数大于 1 意味有加速。从图 9看出，加速倍数在 13.10~87.22 之间，平均值为 38.25。反拉方法在训练集和测试集上正确分别为 99.23% 和 98.82%，相对于未反拉时正确率的提升见图 10，训练集上平均提高 0.51%，测试集上平均提高 0.14%。

5.2 加速 CIFAR10 训练

手写数字数据集 CIFAR10 [13] 包含 10 类彩色图片，图片大小为 32×32，将 2 维彩色图像拉平制作为 1 维向量。取任意两类图片分别作为负样本、正样本进行训练，组合顺序依次为 1-2、1-3、. . .、1-10、2-3、2-4、. . .、2-10、. . .、8-9、9-10，一共 45 种组合。每种组合训练 10 次，然后训练下一种组合，共计训练 450 次。训练使用负梯度下降法，步长指定为 0.0001，无正则化项，$\mathbf{w}$ 的初值从均匀分布 $U(-1/\sqrt{3072}, 1/\sqrt{3072})$ 中随机选取，b 的初值为 0。最大迭代步数设为 20000，最优迭代步数如图 11如示，加速倍数如图 12所示。加速倍数在 1.80~9.06 之间，平均值为 5.61。反拉方法在训练集和测试集上正确率均值分别为 81.69% 和 81.00%，相对于未反拉时正确率的提升见图 10，训练集上平均提高 2.52%，测试集上平均提高 2.06%。图 11~ 图 13中未显示正确率出现大幅震荡的组合。

5.3 控制过拟合

在 3.1节的例子上应用反拉方法，取 $\lambda = -0.9$，法向量初始值为 $\mathbf{w} = (1/\sqrt{2}; -1/\sqrt{2})$，截距 $b = 0$，初始分隔线如图 14中黑线所示。用负梯度下降法迭代求解式 (11)，迭代步长指定

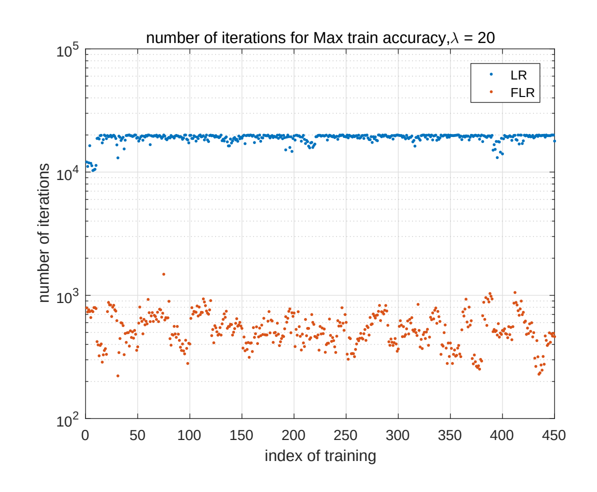


图 8: 在 *MNIST* 上，*LR* 最大正确率对应的训练次数，横轴是各个组合的编号。*LR* 表示未用反拉加速，*FLR* 代表使用了反拉加速。

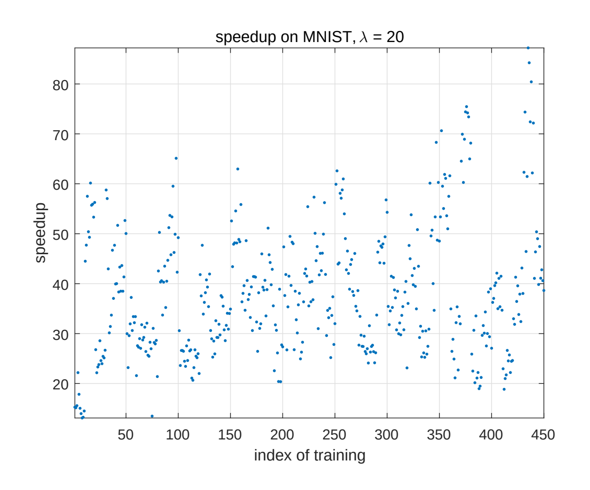


图 9: 在 *MNIST* 上，反拉方法获得的加速倍数。

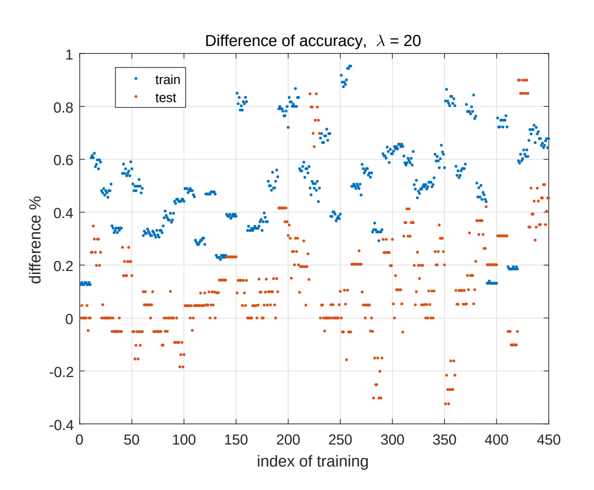


图 10: 在 *MNIST* 上，反拉方法对正确率的影响。

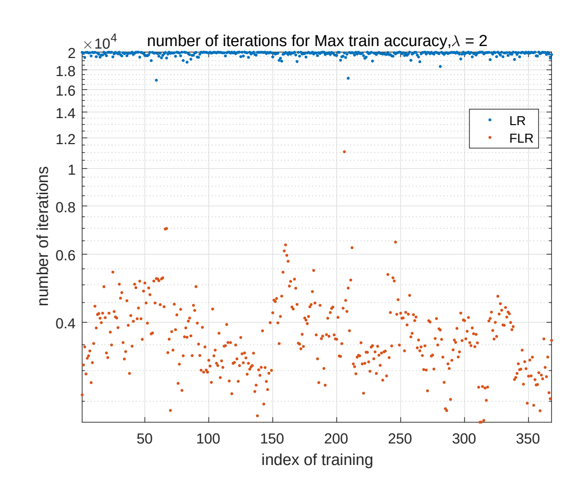


图 11: 在 *CIFAR10* 上，*LR* 最大正确率对应的训练次数，横轴是各个组合的编号。

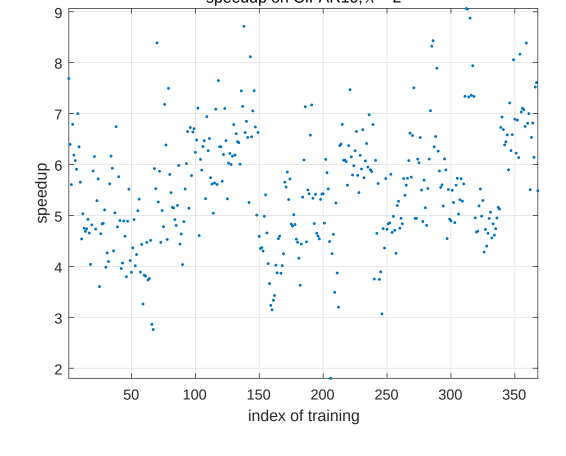


图 12: 在 *CIFAR10* 上，反拉方法获得的加速倍数。

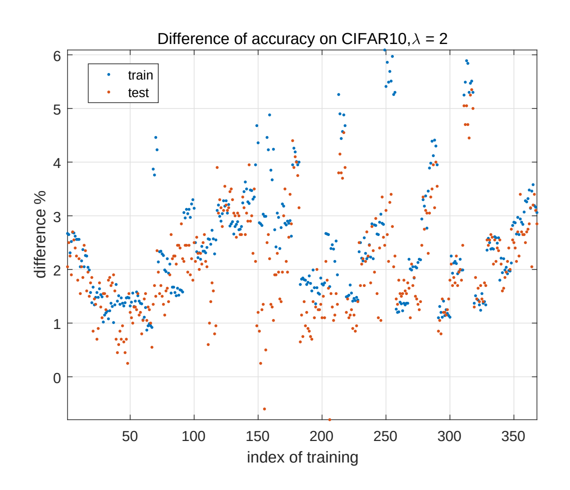


图 13: 在 *CIFAR10* 上，反拉方法对正确率的影响。

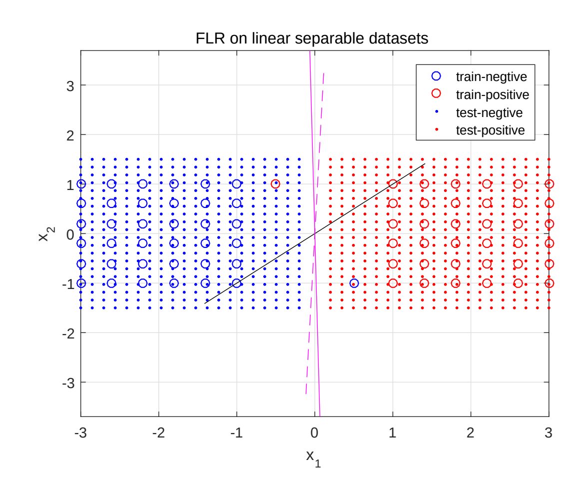


图 14: 线性可分训练集上，$\lambda = -0.9$ 时反拉方法的训练效果。

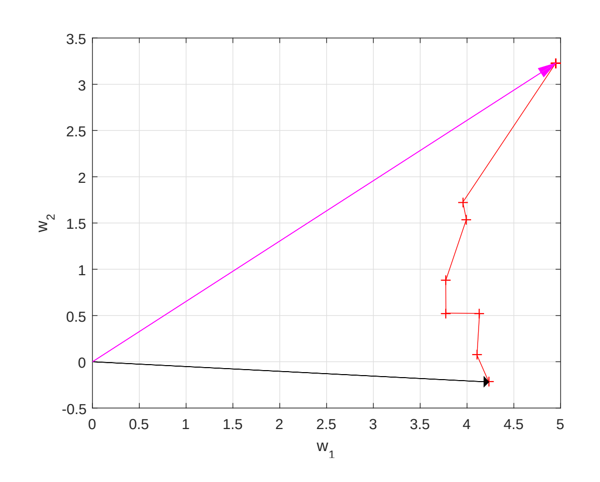


图 15: 在线性可能分训练集上，最优法向量 $\hat{\mathbf{w}}$ 随 λ 的变化情况，右上角的 + 号代表 *13* 个相互接近的点。

为 0.1，前后两步迭代的损失函数值小于 $\epsilon = 10^{-6}$ 时停止迭代。洋红色虚线是迭代 1281 步后的分隔线，洋红色实直线是迭代 2561 步后停止时的最优分隔线。最优分隔线的斜截式方程为 $4.1066x_1+0.0718x_2+1.5237\times10^{-17} = 0$，调整系数后的等价方程为 $x_1+0.0175x_2+3.7105\times10^{-18} = 0$，与人眼观察的理想分隔线 $x_1 = 0$ 很接近。此时，反拉方法有效缓解了过拟合。

在 3.1节的例子上应用反拉方法，λ 在区间 $[-1, 1]$ 上均匀取 21 个值，迭步长指定为 0.1，前后两步迭代的损失函数值小于 $\epsilon = 10^{-6}$ 时停止迭代。将所得的 21 个最优法向量 $\hat{\mathbf{w}}$ 绘制出来，得到图 15。黑色带箭头直线是 $\lambda = -1$ 时得到 $\hat{\mathbf{w}}$，洋红色带箭头直线是 $\lambda = 1$ 时得到 $\hat{\mathbf{w}}$，折线上的 + 号对应 $\lambda \in (-1, 1)$ 时得到的 $\hat{\mathbf{w}}$。注意，这个线性不可分样本集的理想分隔线是 $x_1 = 0$，它的法向量 $\mathbf{w} = (1; 0)$。从图 15知，$\lambda = -0.9$ 时的法向量方向与理想法向量最接近，随着 λ 的增大，最优法向量与理想法向量的夹角越来越大，过拟合起来越严重。这个实验证明，反拉系数 λ 能够控制线性可分数据集上的过拟合。

在 3.2节的例子上应用反拉方法，取 $\lambda = -0.8$，法向量初始值为 $\mathbf{w} = (1/\sqrt{2}; -1/\sqrt{2})$，截距 $b = 0$，初始分隔线如图 16黑线所示。用负梯度下降法迭代求解式 (11)，迭代步长指定为 0.1，前后两步迭代的损失函数值小于 $\epsilon = 10^{-6}$ 时停止迭代。洋红色虚线是迭代 1057 步后的分隔线，洋红色实直线是迭代 2116 步后停止时的最优分隔线。最优分隔线的斜截式方程为 $3.8356x_1 - 0.0140x_2 + 0.0208 = 0$，调整系数后的等价方程为 $x_1 - 0.0037x_2 + 0.0054 = 0$，与人眼观察的理想分隔线 $x_1 = 0$ 很接近。此时，反拉方法有效缓解了过拟合。

在 3.2节的例子上应用反拉方法，λ 在区间 $[-1, 1]$ 上均匀取 21 个值，代步长指定为 0.1，前后两步迭代的损失函数值小于 $\epsilon = 10^{-6}$ 时停止迭代。将所得的 21 个最优法向量 $\hat{\mathbf{w}}$ 绘制出来，得到图 17。黑色带箭头直线是 $\lambda = -1$ 时得到 $\hat{\mathbf{w}}$，洋红色带箭头直线是 $\lambda = 1$ 时得到 $\hat{\mathbf{w}}$，折线上的 + 号对应 $\lambda \in (-1, 1)$ 时得到的 $\hat{\mathbf{w}}$。注意，这个线性不可分样本集的理想分隔线是 $x_1 = 0$，它的法向

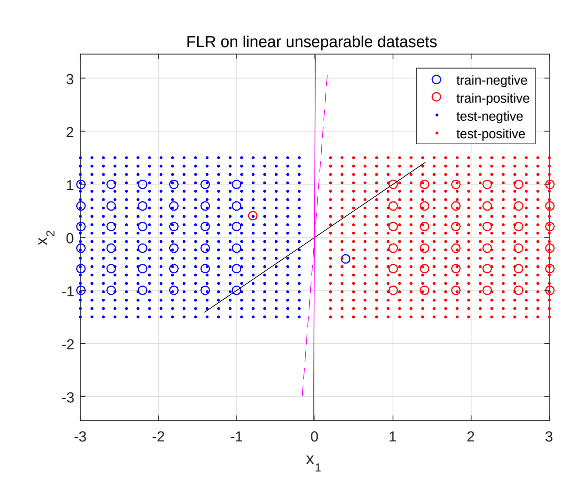


图 16: 线性不可分训练集上，$\lambda = -0.8$ 时反拉方法的训练效果。

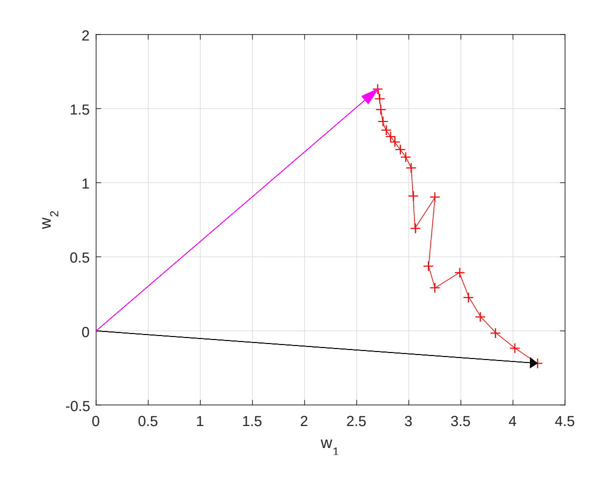


图 17: 在线性不可能分训练集上，最优法向量 $\hat{\mathbf{w}}$ 随 λ 的变化情况。

量 $\mathbf{w} = (1; 0)$。从图 17，$\lambda = -0.8$ 时的法向量方向与理想法向量最接近，随着 λ 的增大，最优法向量与理想法向量的夹角越来越大，过拟合起来越严重。这个实验证明，反拉系数 λ 能够控制线性不可分数据集上的过拟合。

6 总结与展望

本文用严格数学分析来解释逻辑回归过拟合现象，进而得到了加速训练过程的反拉方法和保证交叉熵数值稳定的近似方法。由过拟合原因的推导过程知道，反拉加速会导致更加严重的过拟合，必须采取应对措施。可以添加常规的正则项，也可以将反拉系数逐渐减至 0 以下。根据数值实验经验，反拉系数 λ 过大时，正确率会降低，正确率曲线震荡。在实际应用中，应首先保证正确率曲线平滑，再追求加速性能。

反拉方法的加速效果看起来与样本集有一定的关联，其间的作用机理需要进一步研究。

参考文献

[1] 周志华. 机器学习. 清华大学出版社，2016.4

[2] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification Journal of Machine Learning Research 9(2008), 1871-1874.

[3] Simon Wiesler, Hermann Ney. A Convergence Analysis of Log-Linear Training. Advances in Neural Information Processing Systems , 2011 :657-665

[4] Sebastian Ruder. An overview of gradient descent optimization algorithms. arXiv:1609.04747 [cs.LG]

[5] Brendan H., Holt G., Sculley D., Young M., Ebner D., Grady J., Nie L., Phillips. T, Davydov E., Golovin D., Chikkerur S., Liu D., Wattenberg M., Hrafnkelsson A., Boulos T., Kubica J. (2013) Ad Click Prediction: a View from the Trenches. Proceedings of the 19-th KDD.

[6] G. Andrew and J. Gao. Scalable training of l1-regularized log-linear models. In Proceedings of the 24th international conference on Machine learning, ICML'07, pages 33–40, New York, NY, USA, 2007. ACM.

[7] C.-J. Lin and J. J. Mor e. Newton's method for large bound-constrained optimization problems. SIAM J. on Optimization, 9(4):1100–1127, Apr. 1999.

[8] T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning, Second Edition: Data Mining, Inference, and Prediction. Springer Series in Statistics. Springer, 0002-2009. corr. 3rd edition, Feb. 2009.

[9] P. Domingos. A unified bias-variance decomposition and its applications. In Proceedings of the Seventeenth International Conference on Machine Learning, pages 231–238, Stanford, CA, 2000. Morgan Kaufmann.

[10] Pedro Domingos. A Few Useful Things to Know about Machine Learning. Communications of the ACM, Vol. 55 No. 10, Pages 78-87, 2012.

[11] 何沧平，对焦分类方法，[ChinaXiv:201711.02399]

[12] Yann LeCun, Corinna Cortes, Christopher J.C. Burges. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/

[13] Alex Krizhevsky. The CIFAR-10 dataset. http://www.cs.toronto.edu/ kriz/cifar.html